

Adaptive Quantization for Predicting Transform-based Point Cloud Compression

Xiaohui Wang¹, Guoxia Sun¹(✉), Hui Yuan^{2,3}, Raouf Hamzaoui², and Lu Wang¹

¹ School of Information Science and Engineering, Shandong University, Qingdao, China
sun_guoxia@sdu.edu.cn

² School of Engineering and Sustainable Development, De Montfort University, Leicester, UK

³ School of Control Science and Engineering, Shandong University, Jinan, China

Abstract. The representation of three-dimensional objects with point clouds is attracting increasing interest from researchers and practitioners. Since this representation requires a huge data volume, effective point cloud compression techniques are required. One of the most powerful solutions is the Moving Picture Experts Group geometry-based point cloud compression (G-PCC) emerging standard. In the G-PCC lifting transform coding technique, an adaptive quantization method is used to improve the coding efficiency. Instead of assigning the same quantization step size to all points, the quantization step size is increased according to level of detail traversal order. In this way, the attributes of more important points receive a finer quantization and have a smaller quantization error than the attributes of less important ones. In this paper, we adapt this approach to the G-PCC predicting transform and propose a hardware-friendly weighting method for the adaptive quantization. Experimental results show that compared to the current G-PCC test model, the proposed method can achieve an average Bjøntegaard delta rate of -6.7%, -14.7%, -15.4%, and -10.0% for the luma, chroma Cb, chroma Cr, and reflectance components, respectively on the MPEG Cat1-A, Cat1-B, Cat3-fused and Cat3-frame datasets.

Keywords: point cloud compression, G-PCC, predicting transform, adaptive quantization.

1 Introduction

With the development of 3D scanning technology, point clouds are becoming more and more popular to represent the surface of 3D objects and scenes. Point clouds contain geometry and attribute information. The geometry information is represented by a list of 3D coordinates, while the attribute information may include color, reflectance, normal direction, etc. Point clouds are used in various applications such as gaming, autonomous navigation, virtual reality, cultural heritage, and so on. A point cloud usually contains millions of points, resulting in a huge amount of data. For efficient storage and transmission, point cloud compression [1] is required.

Many point cloud attribute compression methods have been proposed [2]. In this paper, we focus on the Moving Picture Experts Group (MPEG) G-PCC. G-PCC can

be divided into geometry and attribute compression. Geometry compression usually uses an octree to voxelize the original point cloud. In attribute compression, the redundancy is removed by inter-point or inter-block prediction, followed by efficient transform of the prediction residuals, under the guidance of geometry information. Next, quantization is applied to compress the residuals. Generally, the quantization step size is uniform for each coding unit (point or block). The larger the quantization step size, the lower the bitrate (and the greater the reconstruction error), and vice versa. Thus, the quantization step size plays a vital role in compression performance and reconstruction quality [3].

However, a uniform quantization step size only considers the global information and lacks flexibility. For example, because of the inter-point dependency, a smaller quantization step size should be applied to the residual values of the important points as they will affect the reconstruction of their successors in the coding order.

G-PCC offers three solutions for attribute coding: region-adaptive hierarchical transform (RAHT) [4], predicting transform (PT) [5] and lifting transform (LT) [6]. Both PT and LT are LoD-based methods. To improve the coding efficiency of PT, we apply the adaptive quantization method used in LT to PT and propose a hardware-friendly weighting method for the adaptive quantization. Given a voxelized point cloud, assume that the geometry information has been encoded and the LoDs have already been generated. We traverse all points in reverse LoD-based coding order to generate quantization weights for the nearest neighbors of the current point. Points that are more frequently used to predict subsequent points are considered to be more important, and their quantization weights are set to be larger. Then, the prediction residuals are quantized by multiplying them by the ratio of the quantization weight to the quantization step size. Experimental results demonstrate that the proposed method can achieve better compression performance than the latest G-PCC test model [7].

The remainder of the paper is organized as follows. In Section 2, related work is briefly presented. Section 3 describes the details of the proposed adaptive quantization method. Experimental results and conclusions are given in Section 4 and Section 5, respectively.

2 Related Work

Fig. 1 shows the encoder architecture of G-PCC. Using the reconstructed geometry information, the points are recolored by the attribute transfer module. The recolored points are then predicted and transformed by RAHT, PT or LT. Finally, the attribute residuals are quantized and entropy coded by the quantization and arithmetic encoder modules, respectively. Tabatabai *et al.* [8] proposed a conversion method between the quantization parameter (QP) and the quantization step size. Iguchi *et al.* [9] proposed a pre-defined QP table to determine the quantization step size and reduce the conversion processing time. Dean and Iguchi [10] and Dean [11] proposed a quantization control method based on LoDs and used different quantization step sizes for different LoD layers. All these previous quantization methods use a fixed quantization step size for all points in the same LoD layer or in a group of LoD layers. Mammou *et al.* [6]

note that the LoD-based prediction strategy makes points in lower LoDs more influential since they are used more often for prediction and propose to use influence weights computed during the transform process in order to guide the quantization process of LT attribute coding. In this paper, we adapt this approach for PT attribute coding and propose a hardware friendly weighting method for the adaptive quantization. Extensive experimental results show that our method can improve the coding efficiency of PT attribute coding in G-PCC.

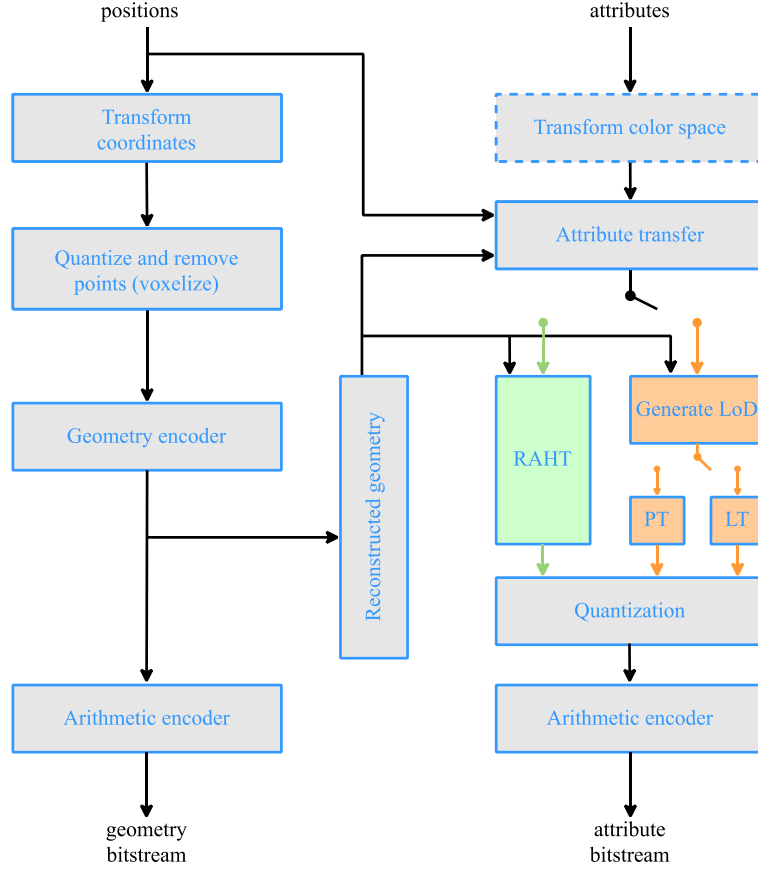


Fig. 1. G-PCC encoder architecture [12]

3 Proposed Method

We first start by explaining how the LoDs are formed. Let L be the number of LoDs and let $(d_l)_{l=1 \dots L-1}$ be a set of user-defined thresholds which satisfy $d_l < d_{l-1}$ and $d_{L-1} = 0$. Then, a point cloud can be partitioned into a set of disjoint subsets called refinement levels $(R_l)_{l=0 \dots L-1}$ as follows [12, 13]. First, add all the points in the point

cloud to the set of non-visited points NV and set the set of visited points V to the empty set. At iteration $l = 0$, remove the first point in NV and add it to both R_0 and V . Next, at each iteration $l \geq 1$, generate a refinement level R_l as follows: iterate over all the points in set NV and compute the minimum distance D of the current point to the set V . If D is greater than or equal to d_l , then the current point is removed from NV and added to both R_l and V ; otherwise skip this point. The level of detail LoD_l is obtained by taking the union of the refinement levels R_0, R_1, \dots, R_l . Fig. 2 shows an example with $L = 4$.

3.1 PT in G-PCC

For PT, the attribute values are encoded/decoded by following the coding order defined in the LoD generation process, i.e., the LoD-based coding order (see the example in Fig. 2). Specifically, the attribute value of each point in $R_j (j > 0)$ is predicted from the attribute values of its three nearest neighbors in LoD_{j-1} where only the already encoded/decoded points are considered for prediction. In the example of Fig. 2, suppose that P_7 in R_2 is the point to be encoded. Then the already encoded points P_0 , P_1 , and P_{11} in LoD_1 will be used for the prediction. After prediction, the prediction residuals (i.e., the difference between original and predicted attribute values) are quantized and entropy coded by an arithmetic encoder. Finally, from the inverse quantized residuals and the predicted values, the attribute information of a point can be reconstructed. This LoD-based prediction strategy makes the points in lower LoD layers more important than those in higher layers since they are used more often for prediction.

G-PCC uses a fixed quantization step size to quantize the prediction residuals of all points. But quantization usually causes distortion, i.e., the reconstructed attributes are not equal to the original ones. Since the distortion of points in lower LoD layers will directly affect the prediction accuracy of the subsequent points, it is beneficial to use a different quantization step size for points with different importance.

3.2 Proposed Adaptive Quantization Method

To apply different quantization steps to points with different importance, we introduce a quantization weight that measures the importance of each point. Let $w(P)$ be the quantization weight of point P . We compute $w(P)$ by applying the following traversal procedure:

Step 1: Set the initial weight of all points to 256.

Step 2: Traverse the points according to the inverse LoD-based coding order (i.e., the order determined by increasing the distance (e.g., Manhattan or Euclidean) to the origin point).

Step 3: For the current point P_i in R_j , update the quantization weights of its three nearest neighbors $(P_{i,k})_{k=1,2,3}$ in LoD_{j-1} as follows:

$$w(P_{i,k}) \leftarrow w(P_{i,k}) + ((\alpha(P_{i,k}) \times w(P_i)) \gg 8), \quad (1)$$

where $P_{i,k}$ is the k -th nearest neighbor of P_i in LoD_{j-1} , and

$$\alpha(P_{i,k}) = 2^{6-k}, \quad (2)$$

Fig. 3 illustrates the computation process of the quantization weights. Since the quantization weights are completely determined by the reconstructed geometry, they do not need to be encoded into the bitstream. Also, since the computation of the quantization weight is achieved by arithmetic shift and integer multiplication, the proposed method is hardware-friendly.

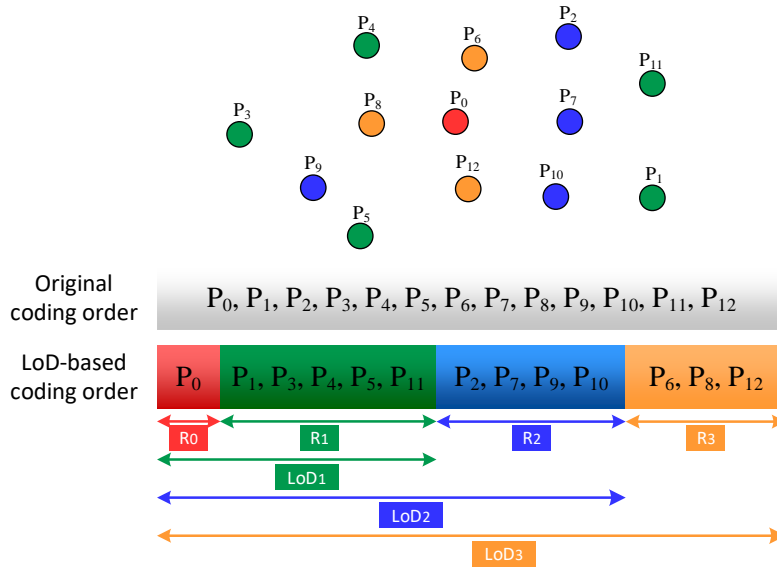


Fig. 2. LoD generation procedure. In the LoD-based coding order, the points are ordered according to decreasing distance (e.g., Manhattan or Euclidean distance) to the origin point (P_0).

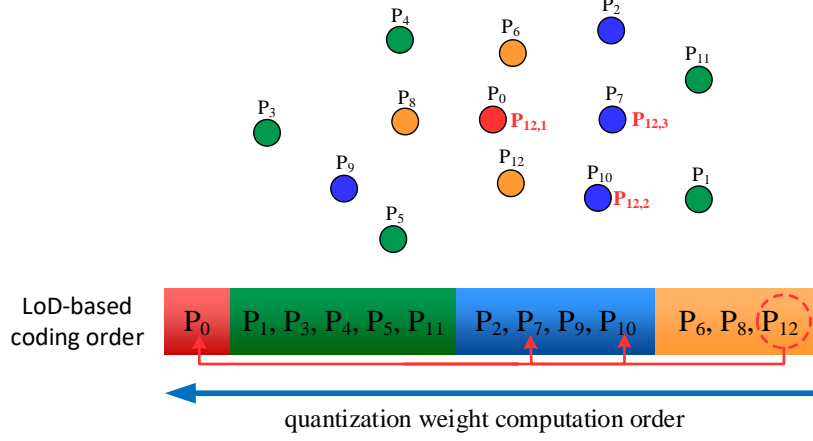


Fig. 3. Quantization weight computation process. The quantization weights of P_0 , P_7 , and P_{10} are computed first.

After updating the quantization weight, the residual of each point is quantized as follows. Let a be the original attribute value of point P_i and \hat{a} be the predicted attribute value. The prediction residual r is given by

$$r = a - \hat{a}. \quad (3)$$

If Q_{step} is the quantization step for P_i , then the quantized residual can be written as

$$\bar{r} = \left\lfloor \frac{a - \hat{a}}{Q_{step}} \right\rfloor = \left\lfloor \frac{r}{Q_{step}} \right\rfloor, \quad (4)$$

where $\lfloor x \rfloor$ is the greatest integer smaller than or equal to x . At the encoder side, the prediction residual associated with P_i is multiplied by the quantization weight $w(P_i)$. Thus, the adaptive quantized residual of P_i can be written as

$$\bar{r}' = \left\lfloor \frac{r \times w(P_i)}{Q_{step}} \right\rfloor = \left\lfloor \frac{r}{Q_{step}/w(P_i)} \right\rfloor, \quad (5)$$

where the actual quantization step of point P_i can be seen as $Q_{step}/w(P_i)$. To make sure that the actual quantization step is greater than or equal to one, we replace (5) by

$$\bar{r}' = \left\lfloor \frac{r}{Q_{step}/\min(Q_{step}, w(P_i))} \right\rfloor. \quad (6)$$

The proposed adaptive quantization method improves the reconstruction quality of the important points in lower LoD layers by applying different quantization steps for points with different importance. In this way, the overall prediction accuracy of the point cloud and the coding efficiency can be enhanced.

4 Experimental Results

We implemented the proposed method in the latest G-PCC reference software (TMC13 version 12.0 [7]) and conducted experiments under the CY condition (lossless geometry, near-lossless attribute) of the Common Test Conditions (CTC) [14] using the MPEG Category 1-A, Category 1-B, Category 3-fused, and Category 3-frame datasets. The G-PCC CTC consist of four test conditions: C1, C2, CW, and CY. C1 is for lossless geometry and lossy attributes applications. C2 is for lossy geometry and lossy attributes applications. CW is for lossless geometry and lossless attributes applications. CY is for lossless geometry and near-lossless attributes applications. The first two test conditions (C1 and C2) adopt LT or RAHT for attribute compression, while the last two (CW and CY) adopt PT for attribute compression. Category 1-A and Category 1-B include static objects and scenes with position and color information. Category 3-fused includes dynamically acquired point clouds of a single frame with position, color and reflectance information. Category 3-frame includes dynamically acquired point clouds with position and reflectance information. Fig. 4 shows an example from each category.

The computer used for conducting the experiments has a 3.70GHz Intel Core i7-7700 processor and 32GB RAM. To evaluate the rate-distortion performance, we computed the distortion at five bitrates obtained by varying the quantization parameters. The distortion was computed with the point-to-point peak signal-to-noise ratio (PSNR) metric proposed by MPEG [15]. Full details of the coding configurations can be found in [7].



Fig. 4. Point clouds from four categories. (a) loot_viewdep_vox12 form Cat1-A (b) ulb_unicorn_hires_vox15 from Cat1-B (c) citytunnel_q1mm from Cat3-fused (d) Ford_01_vox1mm-100 from Cat3-frame

Table 1 shows the experimental results. In the table, Bjøntegaard delta rate (BD-rate) (%) [16] represents the average attribute rate increment of the proposed method compared to TMC13 at the same PSNR, while Complexity Ratio (%) is the ratio of the encoding time (resp. decoding time) between the proposed method and TMC13. We can see that an average -6.5%, -14.5%, -15.2% and -0.1% BD-rate for Luma, Chroma Cb, Chroma Cr and Reflectance can be achieved, respectively. We can also see that the encoding complexity and decoding complexity of the proposed method are increased only slightly. Fig. 5 compares the rate-PSNR curves of TMC13 and the

proposed method. We can see that the proposed method can achieve better rate-distortion performance than TMC13, especially at low bitrates.

Table 1. Average BD-rate for attributes and complexity ratio on CY

| Dataset Category | BD-rate (%) | | | | Complexity Ratio (%) | |
|---------------------|-------------|--------------|--------------|-------------|----------------------|------------|
| | Luma | Cb | Cr | Reflectance | Encoding | Decoding |
| Cat1-A | -10.6 | -21.1 | -22.0 | NA | 102 | 103 |
| Cat1-B | -3.6 | -10.4 | -10.8 | NA | 101 | 102 |
| Cat3-fused | -0.3 | -0.2 | -0.2 | -0.1 | 102 | 102 |
| Cat3-frame | NA | NA | NA | -0.2 | 102 | 102 |
| Overall | -6.5 | -14.5 | -15.2 | -0.1 | 102 | 102 |

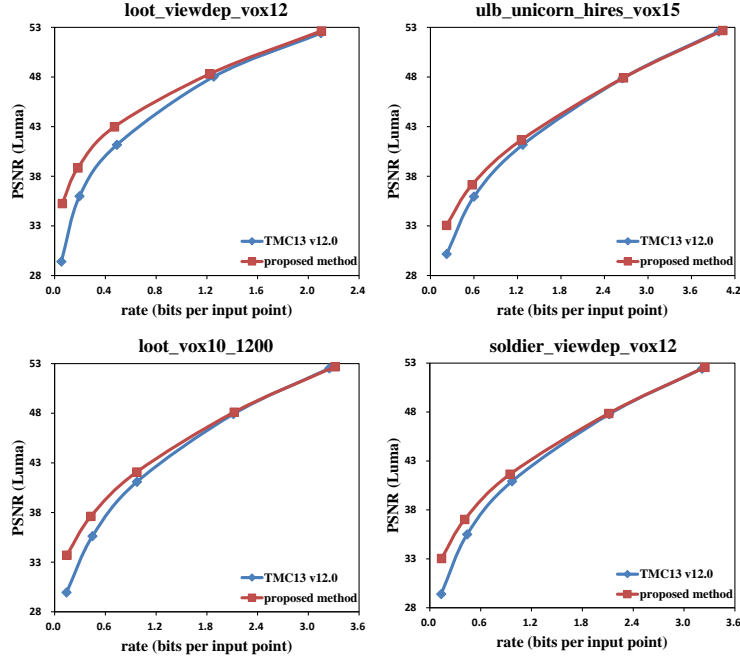


Fig. 5. Rate-PSNR curves. The PSNR is computed for the Luma component and the rate is computed for three color components.

Table 2 shows the detailed BD-rate of each sequence. By analyzing the results, we can see that the average gain for Cat1 is larger than that for Cat3. This is because Cat3 consists of only one LoD layer, and the proposed method is not suitable when the points have similar importance.

Therefore, we forced the number of LoD layers in Cat3 to be equal to 12. Table 3 shows that an average -6.7%, -14.7%, -15.4%, and -10% BD-rate reduction can be achieved by the proposed method.

Table 2. BD-rate of each sequence for attributes on CY.

| Dataset Category | Sequence Name | BD-rate (%) | | | |
|------------------|----------------------------------|-------------|-------|-------|-------------|
| | | Luma | Cb | Cr | Reflectance |
| Cat1-A | basketball_player_vox11_00000200 | -22.9 | -25.7 | -40.3 | NA |
| | boxer_viewdep_vox12 | -24.6 | -46.2 | -47.4 | NA |
| | dancer_vox11_00000001 | -18.2 | -23.7 | -36.6 | NA |
| | egyptian_mask_vox12 | -7.6 | -23.9 | -12.4 | NA |
| | facade_00064_vox11 | -4.1 | -10.2 | -16.4 | NA |
| | frog_00067_vox12 | -6.1 | -13.0 | -14.5 | NA |
| | longdress_viewdep_vox12 | -4.5 | -12.9 | -10.3 | NA |
| | loot_viewdep_vox12 | -28.4 | -56.2 | -54.0 | NA |
| | queen_0200 | -12.5 | -28.8 | -26.2 | NA |
| | redandblack_viewdep_vox12 | -20.9 | -25.4 | -22.0 | NA |
| | shiva_00035_vox12 | 1.3 | -0.7 | -1.0 | NA |
| | soldier_viewdep_vox12 | -15.5 | -34.2 | -40.6 | NA |
| | thaidancer_viewdep_vox12 | -12.2 | -21.5 | -18.5 | NA |
| | ulb_unicorn_vox13 | -8.8 | -19.1 | -18.0 | NA |
| Cat1-B | arco_valentino_dense_vox12 | 1.3 | -0.2 | 0.2 | NA |
| | egyptian_mask_vox20 | -7.5 | -24.5 | -12.2 | NA |
| | facade_00064_vox20 | -4.3 | -11.0 | -18.7 | NA |
| | frog_00067_vox20 | -5.7 | -12.5 | -14.2 | NA |
| | head_00039_vox20 | -2.4 | -9.7 | -14.7 | NA |
| | house_without_roof_00057_vox20 | -5.4 | -10.3 | -11.4 | NA |
| | landscape_00014_vox20 | -0.1 | -7.3 | -7.5 | NA |
| | palazzo_carignano_dense_vox20 | 1.4 | 0.0 | 0.6 | NA |
| | stanford_area_2_vox20 | -8.1 | -13.6 | -15.8 | NA |
| | staue_klimt_vox20 | 0.2 | -2.9 | -3.2 | NA |
| | ulb_unicorn_hires_vox20 | -10.5 | -29.4 | -26.8 | NA |
| | ulb_unicorn_vox20 | -9.3 | -18.9 | -18.3 | NA |
| Cat3-fused | citytunnel_q1mm | -0.3 | -0.2 | -0.2 | -0.3 |
| | overpass_q1mm | -0.1 | -0.1 | -0.1 | 0.0 |
| | tollbooth_q1mm | -0.5 | -0.3 | -0.2 | 0.0 |
| Cat3-frame | ford_01_q1mm | NA | NA | NA | 0.0 |
| | ford_02_q1mm | NA | NA | NA | 0.0 |
| | ford_03_q1mm | NA | NA | NA | 0.0 |
| | qnxadas-junction-approach | NA | NA | NA | -0.3 |
| | qnxadas-junction-exit | NA | NA | NA | -0.1 |
| | qnxadas-motorway-join | NA | NA | NA | -0.3 |
| | qnxadas-navigating-bends | NA | NA | NA | -0.5 |

Table 3. Average BD-rate for attributes and complexity ratio on CY. The number of LoD layers was changed to 12 for the Cat3 dataset.

| Dataset Category | BD-rate (%) | | | | Complexity Ratio (%) | |
|-----------------------------|--------------------|--------------|--------------|--------------|-----------------------------|------------|
| | Luma | Cb | Cr | Reflectance | Encoding | Decoding |
| Cat1-A | -10.6 | -21.1 | -22.0 | NA | 102 | 103 |
| Cat1-B | -3.6 | -10.4 | -10.8 | NA | 101 | 102 |
| Cat3-fused | -2.6 | -3.3 | -3.2 | -1.8 | 102 | 102 |
| Cat3-frame | NA | NA | NA | -13.6 | 102 | 102 |
| Overall | -6.7 | -14.7 | -15.4 | -10.0 | 102 | 102 |

5 Conclusion

To improve the coding efficiency of PT in G-PCC, we adapted the adaptive quantization method used in LT to PT. To take into account the relative importance of the points, we introduced a hardware-friendly quantization weighting method by analyzing the LoD structure. Experimental results show that the proposed method outperforms PT attribute coding with the latest G-PCC test model [7].

Acknowledgement

This work has received funding from the National Natural Science Foundation of China under Grant 61871342, and the OPPO Research Fund.

References

1. Schwarz, S., Preda, M., Baroncini, V., et al.: Emerging MPEG standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9(1), 133–148 (2019).
2. Liu, H., Yuan, H., Liu, Q., et al: A comprehensive study and comparison of core technologies for MPEG 3-D point cloud compression. *IEEE Transactions on Broadcasting* 66(3), 701-717, (2020).
3. Chen, T., Long, C., Su, H., et al.: Layered projection-based quality assessment of 3D point clouds. *IEEE Access*, 9, 88108-88120, (2021).
4. de Queiroz, R. L., Chou, P. A.: Compression of 3D point clouds using a region-adaptive hierarchical transform. *IEEE Transactions on Image Processing* 25(8), 3947-3956, (2016).
5. Mammou, K.: PCC Test Model Category 3 v0. In: 120th MPEG meeting, document N17249, ISO/IEC JTC1/SC29/WG11, China (2017).
6. Mammou, K., Tourapis, A., Kim, J., et al.: Lifting Scheme for Lossy Attribute Encoding in TMC1. In: 122th MPEG meeting, document m42640, ISO/IEC JTC1/SC29/WG11, US (2018).
7. G-PCC Test Model v12 user manual. In: 132th MPEG meeting, document N00005, ISO/IEC JTC1/SC29/WG7, Online, (2020).

8. Tabatabai, A., Graziosi, D., Zaghetto, A.: New Contribution on Quantization Parameter Definition. In: 126th MPEG meeting, document m47507, ISO/IEC JTC1/SC29/WG11, CH (2019).
9. Iguchi, N., Dean, H. C.: Quantization Parameter Table in Attribute Coding. In: 126th MPEG meeting, document m47401, ISO/IEC JTC1/SC29/WG11, CH (2019).
10. Dean, H. C., Iguchi, N.: Delta QP for Layer of Lifting/Predicting Transform and RAHT. In: 126th MPEG meeting, document m47834, ISO/IEC JTC1/SC29/WG11, CH (2019).
11. Dean, H. C.: CE13.16 Report on Slice-based quantization control. In: 126th MPEG meeting, document m47399, ISO/IEC JTC1/SC29/WG11, CH (2019).
12. G-PCC Codec Description. In: 132th MPEG meeting, document N00011, ISO/IEC JTC1/SC29/WG7, Online, (2020).
13. Kathariya, B., Zakharchenko, V., Chen, J., et al.: Binary-tree based Level-of-Details Generation for Attributes Coding in G-PCC. In: 124th MPEG meeting, document m44940, ISO/IEC JTC1/SC29/WG11, China (2018).
14. Common Test Conditions for G-PCC. In: 132th MPEG meeting, document N00032, ISO/IEC JTC1/SC29/WG7, Online, (2020).
15. Current Status on Point Cloud Compression. In: 113th MPEG meeting, document N15869, ISO/IEC JTC1/SC29/WG11, CH, (2015)
16. Bjontegaard, G.: Improvements of the BD-PSNR Model. In: 35th ITU meeting, document VCEG-AI11, ITU-T SG16 Q.6, Germany (2008).